A Cloud Resource Orchestration Framework for Simplifying the Management of Web Applications

Rajiv Ranjan[1,2], Boualem Benatallah[1], and Mingyi Wang[1]

[1] School of Computer Science and Engineering, University of New South Wales
{rajiv,b.benatallah,mingyi.wang1}@unsw.edu.au
[2]Information Engineering Lab, CSIRO ICT Centre, Canberra
{rajiv.ranjan}@csiro.au

Cloud computing paradigm [1] has shifted the computing from physical hardware- and locally managed software-enabled platforms to virtualized cloud-hosted services. Cloud computing assembles large networks of virtualized services: hardware resources (CPU, storage, and network) and software resources (e.g., databases, load-balancers, monitoring systems, etc.). Key issue in exploiting the potential of cloud computing is "*Resource Orchestration*". Resource orchestration process spans across a range of operations from selection, assembly, and deployment of resources to monitoring their run-time performance statistics (e.g. load, availability, throughput, utilization, etc.). The process aims to ensure achievement of fault-tolerant and QoS fulfillment states by resources and applications through adaptive management.

Existing cloud resource orchestration techniques require human familiarity with different types of resources and typically rely on procedural programming in general-purpose or scripting languages. The interaction with resources is mainly performed through low-level APIs and command line interfaces. Given the proliferation of new providers offering resources at different layers (e.g. software-as-a-service, platform-as-a-service, and infrastructure-as-a-service), such orchestration techniques are therefore inadequate to make the cloud resources accessible to a wide variety of users, particularly from non-IT domain.

To improve this situation, an innovative Resource Orchestration Framework (ROF) is presented in this demonstration. The ROF leverages java widget programming, virtualization platforms and Amazon's open-source APIs [2] to enable simplified, intuitive resource orchestration and web application management. The ROF allows users to graphically browse available resources, Virtual Machine (VM) images, storage repository, application hosting environment, and application components. Further, it supports deployment, configuration, and monitoring of resources and applications directly from Java widgets.

The high level architecture, as shown in Fig. 1 (a), consists of two layers: *Programming Layer:* implements the business logic for the interface exposed by widgets. Programming layer is designed based on an extensible software engineering pattern that allows to plug-in service APIs of different clouds. Though our current implementation works only with Amazon EC2 cloud, it can be easily extended to support other clouds. *Widget Layer:* encapsulates user interface components in the form of six principle widgets including: VM, Instance, Storage, Monitor, Application, and Security. All the authentication and authorization credentials (e.g. secret key, SSH key pairs, security group, etc.) are managed by the Security Widget. On the other hand, Storage widget allows users to upload Java Web Application Archives, database

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 248–249, 2012.

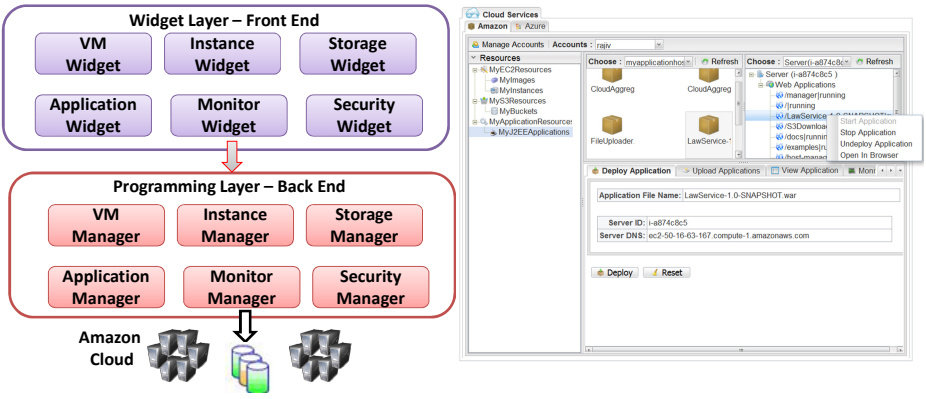

Fig. 1. (a) ROF Architecture. (b) A screenshot of application widget.

schema definition files, and VM images to S3. The basic configuration of VM images including their architecture, imageId, state, and virtualization platform can be viewed through the VM widget. For configuring the deployment (e.g. starting, stopping, and terminating), security credentials, and monitoring preference of VM images, users need to interact with the Instance widget.

The functionality for launching, deploying, and monitoring individual application components and hosting environments is made feasible by Application widget as shown in Fig. 1 (b). This widget also features an inline browser where hosted web application can be actively viewed and tested. The main task of the Monitor widget is to monitor the run-time status of VM instances, network, storage resources and application components. For viewing the performance statistics on per web application basis, one can switch to the Monitor widget.

To our knowledge, no related cloud ROF [2, 3, 4] supports: (i) inline browsing of the hosted web application; (ii) dragging & dropping the application components (e.g. JWARs and HyperSQL [5] DB files) form cloud storage (e.g. S3) to hosting environment (e.g. Tomcat and HyperSQL DB); and (iii) controlling lifecycle activities (e.g. start, stop, refresh, and undeployment) of application components and hosting environments via the widget-level interface.

The purpose of this demonstration is to showcase the effectiveness of ROF in simplifying the process of resource orchestration and web application management. The demonstration will show that prior knowledge of existing cloud resource orchestration tools and concepts is not mandatory for users. Finally, the demonstration will utilize multi-tier web application as the application scenario. It will be deployed within the Tomcat container and its database layer will be managed by HyperSQL [5] database. ROF's screenshots can be found at: http://rranjans.wordpress.com/c-tool.

References

1. Armbrust, M., et al.: A view of Cloud Computing. Communications of the ACM Magazine 53(4), 50–58 (2010), doi:10.1145/1721654.1721672
2. Amazon Web Services, `http://aws.amazon.com/` (accessed September 2011)
3. Rightscale Cloud Framework, `http://www.rightscale.com/` (accessed September 2011)
4. Cloudswitch Framework, `http://www.cloudswitch.com/` (accessed September 2011)
5. HyperSQL Database, `http://hsqldb.org/` (Accessed September 2011)